

# **EXHIBIT B**

UNITED STATES DISTRICT COURT  
NORTHERN DISTRICT OF CALIFORNIA  
SAN FRANCISCO DIVISION

ORACLE AMERICA, INC.

Plaintiff,

v.

GOOGLE INC.

Defendant.

Case No. CV 10-03561 WHA

**(REVISED) OPENING EXPERT REPORT OF ALAN PURDY  
REGARDING COPYRIGHT**

**SUBMITTED ON BEHALF OF PLAINTIFF  
ORACLE AMERICA, INC.**

---

**(Revised)**

**Comparison of Oracle's Java Library API  
Specifications Versus Android's  
Library API Specifications**

**Alan Purdy**  
August 18, 2011

P.O. Box 10666  
Portland OR 97296  
Tel: (503) 274-0784  
Cell: (503) 784-5907

[alan@ili.com](mailto:alan@ili.com)



## Introduction

1. My name is Alan Purdy. I am a Senior Forensic Software Analyst at Johnson-Laird, Inc. (“JLI”).
2. I have been retained as a consultant in this action by counsel for Oracle Corporation.
3. JLI is an Oregon-based corporation that provides consulting services to computer hardware and software manufacturers, and computer-related technical assistance to the legal profession in the United States, Canada, Japan, Singapore, and Europe. JLI specializes in providing consulting services to corporations and attorneys on intellectual property matters (such as forensic analysis of computer-related evidence, copyright and patent infringement, misappropriation of trade secrets, and software due diligence for mergers and acquisitions) and performing assessments of computer software and Techno-archeology™ (*i.e.*, the analysis of software development projects). My *curriculum vitae* is attached as Exhibit 1 to this Report.
4. This Report represents my current opinions based upon the information that I have been supplied—cited herein—and certain analyses that I have been able to perform thus far. I reserve the right to supplement this Report with any additional information provided to me, or if I should be asked to perform additional analyses beyond that which is described below.
5. JLI is compensated at the rate of \$450 per hour for my work in this matter. This compensation is not conditioned on the outcome of this matter.

## Background

6. Oracle provides to Java developers a set of core libraries that include classes of code that perform common functions. Java developers create applications that rely on the Java libraries. The Application Program Interface (“API”) specifications for these libraries are documented on Oracle’s website. (*See, e.g.*, <http://download.oracle.com/javase/1.5.0/docs/api/index.html>). Developers read the API specifications to understand the interfaces to the classes and modules included in the libraries.
7. Google supplies a similar website describing Android’s class libraries for developing programs for Google’s Android platform. (<http://developer.android.com/reference/packages.html>). Google Android’s API specifications currently describe 13 distinct “API Levels” that correspond to different versions of Android.

---

Analysis of Java API Comparing Oracle 5 vs. Android

---

8. I was retained to analyze similarities between the Android API specifications and the Java API specifications. Specifically, my goal was to tabulate, for certain packages within the Java Standard Edition version 5 Development Kit (“JDK 5”), the number of classes, interfaces, and exceptions that are similar or the same in both platforms. I am informed that Oracle is asserting copyright infringement of the 37 Java packages that I was asked to analyze (the “Analyzed Packages”).<sup>1</sup> Additionally, for each class in the Analyzed Packages, I was asked to determine the number of fields, constructors, and method signatures that were the same in both.

9. I was asked to perform comparisons between the following versions of the Java Standard Edition and Android API specifications:

JDK 5 vs. Android API Level 3 (“Cupcake”)

JDK 5 vs. Android API Level 8 (“Froyo”)

I understand that for developers to build Android applications, Google requires them to download the JDK version 5 or 6. (See <http://developer.android.com/sdk/requirements.html>).

## Analysis Methods

10. To perform the requested task, I wrote a Java-based program to extract information from both Oracle’s Java JDK API descriptions, and Google’s Android SDK library descriptions. More particularly, this program started from the list of Analyzed Packages, reading each corresponding package’s web page, and transitively reading all the pages that describe the referenced classes, interfaces, enums, errors, exceptions, and annotation types. Upon visiting each page, the program then captured relevant information for the next steps of the analysis. For example, after reading a page for a specific class, it read and captured the constructors, fields, and methods described on that page. After capturing that information (e.g., such as method signatures), it then wrote the information to a Microsoft Access database for further analysis.

11. After capturing all the necessary information, I then wrote a set of scripts to compare, for example, method, constructor, and field signatures. This document reports on the results of those scripts.

---

<sup>1</sup> I understand that Oracle is not asserting copyright infringement of certain other packages, in some cases because Oracle either uses them under license from third parties or because Oracle allows third parties to utilize these packages under permissive terms. These packages include: java.math, java.util.concurrent, java.util.concurrent.atomic, java.util.concurrent.locks, javax.xml, javax.xml.datatype, javax.xml.namespace, javax.xml.parsers, javax.xml.transform, javax.xml.transform.dom, javax.xml.transform.sax, javax.xml.transform.stream, javax.xml.validation, and javax.xml.xpath. These packages are not included in my analysis.

## Results of Analysis

12. This section presents the results of my analysis in two sections: Analyzed Packages and Analyzed Classes.

### *Analyzed Packages: Results of Analysis*

#### **Packages: Java JDK 5 vs. Android API Level 3**

13. My analysis determined the following metrics with respect to Java JDK 5 libraries and Android's API Level 3 packages:

Package Elements Under Analysis	Java 5 Count	Android API Level 3 Count	In Java 5 but not Android 3 Count (%)	In Android 3 but not Java 5 Count	In Android 3 <u>and</u> Java 5 Count (%)
Packages	37	37	0	0	37 (100%)
Annotation Types	7	0	7 (100%)	0	0 (0%)
Classes	506	458	48 (9.4%)	0	458 (90.6%)
Enums	9	9	0 (0%)	0	9 (100%)
Errors	25	25	0 (0%)	0	25 (100%)
Exceptions	176	165	11 (6.25%)	0	165 (93.75%)
Interfaces	171	158	13 (7.6%)	0	158 (92.4%)
<b>Totals</b>	<b>931</b>	<b>852</b>	<b>79</b>	<b>0</b>	<b>852</b>

#### **Packages: Java JDK 5 vs. Android API Level 8**

14. My analysis determined the following metrics with respect to Java JDK 5 libraries and Android's API Level 8 packages:

Package Elements Under Analysis	Java 5 Count	Android API Level 8 Count	In Java 5 but not Android 8 Count (%)	In Android 8 but not Java 5 Count	In Android 8 <u>and</u> Java 5 Count (%)
Packages	37	37	0	0	37 (100%)
Annotation Types	7	0	7 (100%)	0	0 (0%)
Classes	509	458	48 (9.4%)	0	458 (90.6%)
Enums	9	9	0 (0%)	0	9 (100%)
Errors	25	25	0 (0%)	0	25 (100%)
Exceptions	176	165	11 (6.25%)	0	165 (93.75%)
Interfaces	171	158	13 (7.6%)	0	158 (92.4%)
<b>Totals</b>	<b>934</b>	<b>852</b>	<b>86</b>	<b>0</b>	<b>852</b>

## Analyzed Classes: Results of Analysis

### Classes: Java JDK 5 vs. Android API Level 3

15. My analysis determined the following metrics with respect to Java JDK 5 libraries and Android's API Level 3 classes. Because, within the Analyzed Packages, Android API Level 3 does not provide 48 of Java 5's classes, the table below only compares the fields, constructors and methods ("Class Elements") of those 458 classes in both libraries. The "Java 5 Count" column below therefore identifies the total number of fields (constructors, methods) that exist in Java 5 classes where that class also exists in Android API Level 3 (for the Analyzed Packages). For the right-most column, I compared the signatures of each Class Element, including the modifiers, names, parameters and types, extends (superclass), and implements (interfaces) between Java 5 and Android 3.

Class Elements Compared	Java 5 Count	Android API Level 3 Count	Java 5 Element Matches Android 3 Count
Fields	1062	927	898
Constructors	745	639	438
Methods	5080	5052	3357
<b>Totals</b>	<b>6887</b>	<b>6618</b>	<b>4693</b>

16. When compared, the numbers of methods in each class for corresponding classes between Java 5 and Android API Level 3 have some differences. The following table has three columns, the left-most identifies a class name in which that Java 5 class has a different number of methods than the same class in Android API Level 3. The middle column shows the number of Android 3's methods in that row's class (*see* the left-most column). The right-most column shows the number of Java 5 methods in that row's class. All other classes (of the 458) shared by both systems have the same number of methods in the corresponding class.

Classes With Different Number of Methods Per Class			
Name	Android 3 Method Count	Java 5 Method Count	Difference (Abs)
java.io.BufferedOutputStream	4	3	1
java.io.FileDescriptor	3	2	1
java.io.RandomAccessFile	40	39	1
java.lang.Class	53	57	4
java.lang.Enum	10	9	1
java.lang.Package	17	18	1
java.lang.reflect.Array	20	21	1
java.lang.reflect.Constructor	17	18	1
java.lang.reflect.Field	30	31	1



## Analysis of Java API Comparing Oracle 5 vs. Android

Classes With Different Number of Methods Per Class			
Name	Android 3 Method Count	Java 5 Method Count	Difference (Abs)
java.lang.reflect.Method	21	22	1
java.net.HttpURLConnection	16	17	1
java.net.Inet4Address	10	14	4
java.net.Inet6Address	16	19	3
java.nio.Buffer	14	13	1
java.nio.ByteBuffer	56	57	1
java.nio.DoubleBuffer	24	25	1
java.nio.FloatBuffer	24	25	1
java.nio.IntBuffer	24	25	1
java.nio.LongBuffer	24	25	1
java.nio.ShortBuffer	24	25	1
java.text.BreakIterator	22	25	3
java.text.CollationKey	3	5	2
java.text.DateFormat.Field	2	3	1
java.text.DateFormatSymbols	20	19	1
java.text.DecimalFormatSymbols	32	31	1
java.util.ArrayList	23	20	3
java.util.IdentityHashMap	14	15	1
java.util.jar.JarFile	6	5	1
java.util.regex.Pattern	11	10	1
java.util.TreeMap	16	18	2
java.util.zip.DeflaterOutputStream	6	5	1
java.util.zip.ZipInputStream	6	7	1
javax.security.auth.Subject	12	16	4
<b>TOTAL</b>	<b>620</b>	<b>644</b>	<b>50</b>

**Classes: Java JDK 5 vs. Android API Level 8**


17. My analysis determined the following metrics with respect to Java JDK 5 libraries and Android's API Level 8 classes. Because Android API Level 8 does not provide 48 of Java 5's classes, the table below only compares the fields, constructors and methods ("Class Elements") of those 458 classes from the Analyzed Packages in both libraries. The "Java 5 Count" column below therefore identifies the total number of fields (constructors, methods) that exist in Java 5 classes where that class also exists in Android API Level 8 (for the Analyzed Packages). For the right-most column, I compared the signatures of each Class Element, including the modifiers, names, parameters and types, extends (superclass), and implements (interfaces) between Java 5 and Android 8.

Class Elements Under Analysis	Java 5 Count	Android API Level 8 Count	Java 5 Element Matches Android 8 Count
Fields	1062	927	898
Constructors	745	639	438
Methods	5080	5052	3357
<b>Total</b>	<b>6887</b>	<b>6618</b>	<b>4693</b>

Analysis of Java API Comparing Oracle 5 vs. Android

---

Submitted on Augst 18, 2011, in Portland, Oregon.

---

Alan Purdy

## **Appendix A: Analyzed Packages**

java.awt.font  
java.beans  
java.io  
java.lang  
java.lang.annotation  
java.lang.ref  
java.lang.reflect  
java.net  
java.nio  
java.nio.channels  
java.nio.channels.spi  
java.nio.charset  
java.nio.charset.spi  
java.security  
java.security.acl  
java.security.cert  
java.security.interfaces  
java.security.spec  
java.sql  
java.text  
java.util  
java.util.jar  
java.util.logging  
java.util.prefs  
java.util.regex  
java.util.zip  
javax.crypto  
javax.crypto.interfaces  
javax.crypto.spec

javax.net

javax.net.ssl

javax.security.auth

javax.security.auth.callback

javax.security.auth.login

javax.security.auth.x500

javax.security.cert

javax.sql